

Undecidability and Intractability in Theoretical Physics

Stephen Wolfram

The Institute for Advanced Study, Princeton, New Jersey 08540

(Received 26 October 1984)

Physical processes are viewed as computations, and the difficulty of answering questions about them is characterized in terms of the difficulty of performing the corresponding computations. Cellular automata are used to provide explicit examples of various formally undecidable and computationally intractable problems. It is suggested that such problems are common in physical models, and some other potential examples are discussed.

PACS numbers: 02.90.+p, 01.70.+w, 05.90.+m

There is a close correspondence between physical processes and computations. On one hand, theoretical models describe physical processes by computations that transform initial data according to algorithms representing physical laws. And on the other hand, computers themselves are physical systems, obeying physical laws. This paper explores some fundamental consequences of this correspondence.¹

The behavior of a physical system may always be calculated by simulating explicitly each step in its evolution. Much of theoretical physics has, however, been concerned with devising shorter methods of calculation that reproduce the outcome without tracing each step. Such shortcuts can be made if the computations used in the calculation are more sophisticated than those that the physical system can itself perform. Any computations must, however, be carried out on a computer. But the computer is itself an example of a physical system. And it can determine the outcome of its own evolution only by explicitly following it through: No shortcut is possible. Such computational irreducibility occurs whenever a physical system can act as a computer. The behavior of the system can be found only by direct simulation or observation: No general predictive procedure is possible. Computational irreducibility is common among the systems investigated in mathematics and computation theory.² This paper suggests that it is also common in theoretical physics. Computational reducibility may well be the exception rather than the rule: Most physical questions may be answerable only through irreducible amounts of computation. Those that concern idealized limits of infinite time, volume, or numerical precision can require arbitrarily long computations, and so be formally undecidable.

A diverse set of systems are known to be equivalent in their computational capabilities, in that particular forms of one system can emulate any of the others. Standard digital computers are one example of such "universal computers": With fixed intrinsic instructions, different initial states or programs can be devised to simulate different systems. Some other examples are Turing machines, string transformation systems, recursively defined functions, and Diophan-

tine equations.² One expects in fact that universal computers are as powerful in their computational capabilities as any physically realizable system can be, so that they can simulate any physical system.³ This is the case if in all physical systems there is a finite density of information, which can be transmitted only at a finite rate in a finite-dimensional space.⁴ No physically implementable procedure could then short cut a computationally irreducible process.

Different physically realizable universal computers appear to require the same order of magnitude times and information storage capacities to solve particular classes of finite problems.⁵ One computer may be constructed so that in a single step it carries out the equivalent of two steps on another computer. However, when the amount of information n specifying an instance of a problem becomes large, different computers use resources that differ only by polynomials in n . One may then distinguish several classes of problems.⁶ The first, denoted P , are those such as arithmetical ones taking a time polynomial in n . The second, denoted $PSPACE$, are those that can be solved with polynomial storage capacity, but may require exponential time, and so are in practice effectively intractable. Certain problems are "complete" with respect to $PSPACE$, so that particular instances of them correspond to arbitrary $PSPACE$ problems. Solutions to these problems mimic the operation of a universal computer with bounded storage capacity: A computer that solves $PSPACE$ -complete problems for any n must be universal. Many mathematical problems are $PSPACE$ -complete.⁶ (An example is whether one can always win from a given position in chess.) And since there is no evidence to the contrary, it is widely conjectured that $PSPACE \neq P$, so that $PSPACE$ -complete problems cannot be solved in polynomial time. A final class of problems, denoted NP , consist in identifying, among an exponentially large collection of objects, those with some particular, easily testable property. An example would be to find an n -digit integer that divides a given $2n$ -digit number exactly. A particular candidate divisor, guessed nondeterministically, can be tested in polynomial time, but a systematic solution may require almost all $O(2^n)$ possible candidates to be

tested. A computer that could follow arbitrarily many computational paths in parallel could solve such problems in polynomial time. For actual computers that allow only boundedly many paths, it is suspected that no general polynomial time solution is possible.⁵ Nevertheless, in the infinite time limit, parallel paths are irrelevant, and a computer that solves *NP*-complete problems is equivalent to other universal computers.⁶

The structure of a system need not be complicated for its behavior to be highly complex, corresponding to a complicated computation. Computational irreducibility may thus be widespread even among systems with simple construction. Cellular automata (CA)⁷ provide an example. A CA consists of a lattice of sites, each with k possible values, and each updated in time steps by a deterministic rule depending on a neighborhood of R sites. CA serve as discrete approximations to partial differential equations, and provide models for a wide variety of natural systems. Figure 1 shows typical examples of their behavior. Some rules give periodic patterns, and the outcome after many steps can be predicted without following each intermediate step. Many rules, however, give complex patterns for which no predictive procedure is evident. Some CA are in fact known to be capable of universal computation, so that their evolution must be computationally irreducible. The simplest cases proved have $k=18$ and $R=3$ in one dimension,⁸ or $k=2$ and $R=5$ in two dimensions.⁹ It is strongly suspected that "class-4" CA are generically capable of universal computation: There are such CA with $k=3$, $R=3$ and $k=2$, $R=5$ in one dimension.¹⁰

Computationally, irreducibility may occur in systems that are not full universal computers. For inability to perform, specific computations need not allow all computations to be short cut. Though class-3 CA and other chaotic systems may not be universal computers, most of them are expected to be computationally irreducible, so that the solution of problems concerning their behavior requires irreducible amounts of computation.

As a first example consider finding the value of a site in a CA after t steps of evolution from a finite initial seed, as illustrated in Fig. 1. The problem is specified by giving the seed and the CA rule, together with the $\log t$ digits of t . In simple cases such as the first two shown in Fig. 1, it can be solved in the time $O(\log t)$

necessary to input this specification. However, the evolution of a universal computer CA for a polynomial in t steps can implement any computation of length t . As a consequence, its evolution is computationally irreducible, and its outcome found only by an explicit simulation with length $O(t)$: exponentially longer than for the first two in Fig. 1.

One may ask whether the pattern generated by evolution with a CA rule from a particular seed will grow forever, or will eventually die out.¹¹ If the evolution is computationally irreducible, then an arbitrarily long computation may be needed to answer this question. One may determine by explicit simulation whether the pattern dies out after any specified number of steps, but there is no upper bound on the time needed to find out its ultimate fate.¹² Simple criteria may be given for particular cases, but computational irreducibility implies that no shortcut is possible in general. The infinite-time limiting behavior is formally undecidable: No finite mathematical or computational process can reproduce the infinite CA evolution.

The fate of a pattern in a CA with a finite total number of sites N can always be determined in at most k^N steps. However, if the CA is a universal computer, then the problem is *PSPACE*-complete, and so presumably cannot be solved in a time polynomial in N .¹³

One may consider CA evolution not only from finite seeds, but also from initial states with all infinitely many sites chosen arbitrarily. The value $a^{(t)}$ of a site after many time steps t then in general depends on $2\lambda t \leq Rt$ initial site values, where λ is the rate of information transmission (essentially Lyapunov exponent) in the CA.⁹ In class-1 and -2 CA, information remains localized, so that $\lambda=0$, and $a^{(t)}$ can be found by a length $O(\log t)$ computation. For class-3 and -4 CA, however, $\lambda > 0$, and $a^{(t)}$ requires an $O(t)$ computation.¹⁴

The global dynamics of CA are determined by the possible states reached in their evolution. To characterize such states one may ask whether a particular string of n site values can be generated after evolution for t steps from any (length $n+2\lambda t$) initial string. Since candidate initial strings can be tested in $O(t)$ time, this problem is in the class *NP*. When the CA is a universal computer, the problem is in general *NP*-complete, and can presumably be answered essentially only by testing all $O(k^{n+2\lambda t})$ candidate initial

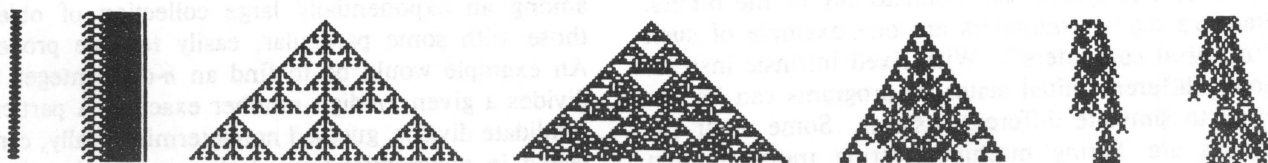


FIG. 1. Seven examples of patterns generated by repeated application of various simple cellular automaton rules. The last four are probably computationally irreducible, and can be found only by direct simulation.

strings.¹⁵ In the limit $t \rightarrow \infty$, it is in general undecidable whether particular strings can appear.¹⁶ As a consequence, the entropy or dimension of the limiting set of CA configurations is in general not finitely computable.

Formal languages describe sets of states generated by CA.¹⁷ The set that appears after t steps in the evolution of a one-dimensional CA forms a regular formal language: each possible state corresponds to a path through a graph with $\Xi^{(t)} < 2^{k^R t}$ nodes. If, indeed, the length of computation to determine whether a string can occur increases exponentially with t for computationally irreducible CA, then the "regular language complexity" $\Xi^{(t)}$ should also increase exponentially, in agreement with empirical data on certain class-3 CA,¹⁷ and reflecting the "irreducible computational work" achieved by their evolution.

Irreducible computations may be required not only to determine the outcome of evolution through time, but also to find possible arrangements of a system in space. For example, whether an $x \times x$ patch of site values occurs after just one step in a two-dimensional CA is in general NP-complete.¹⁸ To determine whether there is any complete infinite configuration that satisfies a particular predicate (such as being invariant under the CA rule) is in general undecidable¹⁸. It is equivalent to finding the infinite-time behavior of a universal computer that lays down each row on the lattice in turn.

There are many physical systems in which it is known to be possible to construct universal computers. Apart from those modeled by CA, some examples are electric circuits, hard-sphere gases with obstructions, and networks of chemical reactions.¹⁹ The evolution of these systems is in general computationally irreducible, and so suffers from undecidable and intractable problems. Nevertheless, the constructions used to find universal computers in these systems are arcane, and if computationally complex problems occurred only there, they would be rare. It is the thesis of this paper that such problems are in fact common.²⁰ Certainly there are many systems whose properties are in practice studied only by explicit simulation or exhaustive search: Few computational shortcuts (often stated in terms of invariant quantities) are known.

Many complex or chaotic dynamical systems are expected to be computationally irreducible, and their behavior effectively found only by explicit simulation. Just as it is undecidable whether a particular initial state in a CA leads to unbounded growth, to self-replication, or has some other outcome, so it may be undecidable whether a particular solution to a differential equation (studied say with symbolic dynamics) even enters a certain region of phase space, and whether, say, a certain n -body system is ultimately stable. Similarly, the existence of an attractor, say,

with a dimension above some value, may be undecidable.

Computationally complex problems can arise in finding eigenvalues or extremal states in physical systems. The minimum energy conformation for a polymer is in general NP-complete with respect to its length.²¹ Finding a configuration below a specified energy in a spin-glass with particular couplings is similarly NP-complete.²² Whenever the stationary state of a physical system such as this can be found only by lengthy computation, the dynamic physical processes that lead to it must take a correspondingly long time.⁵

Global properties of some models for physical systems may be undecidable in the infinite-size limit (like those for two-dimensional CA). An example is whether a particular generalized Ising model (or stochastic multidimensional CA²³) exhibits a phase transition.

Quantum and statistical mechanics involve sums over possibly infinite sets of configurations in systems. To derive finite formulas one must use finite specifications for these sets. But it may be undecidable whether two finite specifications yield equivalent configurations. So, for example, it is undecidable whether two finitely specified four-manifolds or solutions to the Einstein equations are equivalent (under coordinate reparametrization).²⁴ A theoretical model may be considered as a finite specification of the possible behavior of a system. One may ask for example whether the consequences of two models are identical in all circumstances, so that the models are equivalent. If the models involve computations more complicated than those that can be carried out by a computer with a fixed finite number of states (regular language), this question is in general undecidable. Similarly, it is undecidable what is the simplest such model that describes a given set of empirical data.²⁵

This paper has suggested that many physical systems are computationally irreducible, so that their own evolution is effectively the most efficient procedure for determining their future. As a consequence, many questions about these systems can be answered only by very lengthy or potentially infinite computations. But some questions answerable by simpler computations may still be formulated.

This work was supported in part by the U. S. Office of Naval Research under Contract No. N00014-80-C-0657. I am grateful for discussions with many people, particularly C. Bennett, G. Chaitin, R. Feynman, E. Fredkin, D. Hillis, L. Hurd, J. Milnor, N. Packard, M. Perry, R. Shaw, K. Steiglitz, W. Thurston, and L. Yaffe.

¹For a more informal exposition see: S. Wolfram, *Sci. Am.* **251**, 188 (1984). A fuller treatment will be given else-

where.

²E.g., *The Undecidable: Basic Papers on Undecidable Propositions, Unsolvability Problems, and Computable Functions*, edited by M. Davis (Raven, New York, 1965), or J. Hopcroft and J. Ullman, *Introduction to Automata Theory, Languages, and Computations* (Addison-Wesley, Reading, Mass., 1979).

³This is a physical form of the Church-Turing hypothesis. Mathematically conceivable systems of greater power can be obtained by including tables of answers to questions insoluble for these universal computers.

⁴Real-number parameters in classical physics allow infinite information density. Nevertheless, even in classical physics, the finiteness of experimental arrangements and measurements, implemented as coarse graining in statistical mechanics, implies finite information input and output. In relativistic quantum field theory, finite density of information (or quantum states) is evident for free fields bounded in phase space [e.g., J. Bekenstein, *Phys. Rev. D* **30**, 1669 (1984)]. It is less clear for interacting fields, except if space-time is ultimately discrete [but cf. B. Simon, *Functional Integration and Quantum Physics* (Academic, New York, 1979), Sec. III.9]. A finite information transmission rate is implied by relativistic causality and the manifold structure of space-time.

⁵It is just possible, however, that the parallelism of the path integral may allow quantum mechanical systems to solve any NP problem in polynomial time.

⁶M. Garey and D. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness* (Freeman, San Francisco, 1979).

⁷See S. Wolfram, *Nature* **311**, 419 (1984); *Cellular Automata*, edited by D. Farmer, T. Toffoli, and S. Wolfram, *Physica* **10D**, Nos. 1 and 2 (1984), and references therein.

⁸A. R. Smith, *J. Assoc. Comput. Mach.* **18**, 331 (1971).

⁹E. R. Banks, Massachusetts Institute of Technology Report No. TR-81, 1971 (unpublished). The "Game of Life," discussed in E. R. Berlekamp, J. H. Conway, and R. K. Guy, *Winning Ways for Your Mathematical Plays* (Academic, New York, 1982), is an example with $k = 2$, $R = 9$. N. Margolus, *Physica (Utrecht)* **10D**, 81 (1984), gives a reversible example.

¹⁰S. Wolfram, *Physica (Utrecht)* **10D**, 1 (1984), and to be published.

¹¹This is analogous to the problem of whether a computer run with particular input will ever reach a "halt" state.

¹²The number of steps to check ("busy-beaver function") in general grows with the seed size faster than any finite formula can describe (Ref. 2).

¹³Cf. C. Bennett, to be published.

¹⁴Cf. B. Eckhardt, J. Ford, and F. Vivaldi, *Physica (Utrecht)* **13D**, 339 (1984).

¹⁵The question is a generalization of whether there exists an assignment of values to sites such that the logical expression corresponding to the t -step CA mapping is true (cf. V. Sewelson, private communication).

¹⁶L. Hurd, to be published.

¹⁷S. Wolfram, *Commun. Math. Phys.* **96**, 15 (1984).

¹⁸N. Packard and S. Wolfram, to be published. The equivalent problem of covering a plane with a given set of tiles is considered in R. Robinson, *Invent. Math.* **12**, 177 (1971).

¹⁹E.g., C. Bennett, *Int. J. Theor. Phys.* **21**, 905 (1982); E. Fredkin and T. Toffoli, *Int. J. Theor. Phys.* **21**, 219 (1982); A. Vergis, K. Steiglitz, and B. Dickinson, "The Complexity of Analog Computation" (unpublished).

²⁰Conventional computation theory primarily concerns possibilities, not probabilities. There are nevertheless some problems for which almost all instances are known to be of equivalent difficulty. But other problems are known to be much easier on average than in the worst case. In addition, for some NP-complete problems the density of candidate solutions close to the actual one is very large, so approximate solutions can easily be found [S. Kirkpatrick, C. Gelatt, and M. Vecchi, *Science* **220**, 671 (1983)].

²¹Compare *Time Warps, String Edits, and Macromolecules*, edited by D. Sankoff and J. Kruskal (Addison-Wesley, Reading, Mass., 1983).

²²F. Barahona, *J. Phys. A* **13**, 3241 (1982).

²³E. Domany and W. Kinzel, *Phys. Rev. Lett.* **53**, 311 (1984).

²⁴See W. Haken, in *Word Problems*, edited by W. W. Boone, F. B. Cannonito, and R. C. Lyndon (North-Holland, Amsterdam, 1973).

²⁵G. Chaitin, *Sci. Am.* **232**, 47 (1975), and *IBM J. Res. Dev.* **21**, 350 (1977); R. Shaw, to be published.